

DEVICE AND SYSTEM FOR PROCESSING IMAGE

Publication number: JP8044678 (A)

Publication date: 1996-02-16

Inventor(s): TANAKA SHINICHI; KAWAI TOMOAKI; NORO HIDEO +

Applicant(s): CANON KK +

Classification:

- international: G06F15/16; G06F15/177; G06F15/80; G06F9/46; G06F9/50; G06T1/20; G06T7/60; G06T7/00; G06F15/16; G06F15/76; G06F9/46; G06T1/20; G06T7/60; G06T7/00; (IPC1-7): G06F15/16; G06F9/46; G06T1/20; G06T7/00

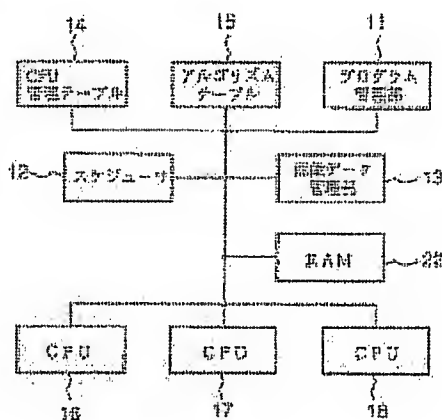
- European:

Application number: JP19940177953 1994072 9

Priority number(s): JP19940177953 1994072 9

Abstract of JP 8044678 (A)

PURPOSE:To dividedly process image data so as to most improve processing efficiency as a whole even on any condition that the loads of respective CPU are dynamically changed or the performance of each CPU is different by dividing all the image data based on the ratio of predictive efficiency of respective CPU and allocating those data to the respective CPU. CONSTITUTION:A CPU managing table 14 manages the performance and present load conditions of CPU 16 to 18 so that the predictive efficiency of respective CPU 16 to 18 can be calculated. Corresponding to the result, a scheduler 12 distributes the image data divided by an image data managing part 13 to the respective CPU 16 to 18, namely, processing is respectively dividedly distributed so that the processing efficiency can be maximized as a whole. At the scheduler 12, the dividing method of image data at respective functions to be used for image processing and the performance of respective CPU 16 to 18 or the like are respectively investigated from an image processing algorithm table and the CPU managing table 14 and utilized as the information for deciding a schedule.



Data supplied from the *espacenet* database — Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-44678

(43) 公開日 平成8年(1996)2月16日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/16	3 8 0 Z			
9/46	3 4 0 D	7737-5B		
	3 6 0 B	7737-5B		
		9061-5H		
			G 0 6 F 15/ 66	K
			15/ 70	3 3 0 F
			審査請求 未請求 請求項の数15	OL (全 12 頁) 最終頁に続く

(21) 出願番号 特願平6-177953

(22) 出願日 平成6年(1994)7月29日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 田中 伸一

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(72) 発明者 河合 智明

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(72) 発明者 野呂 英生

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

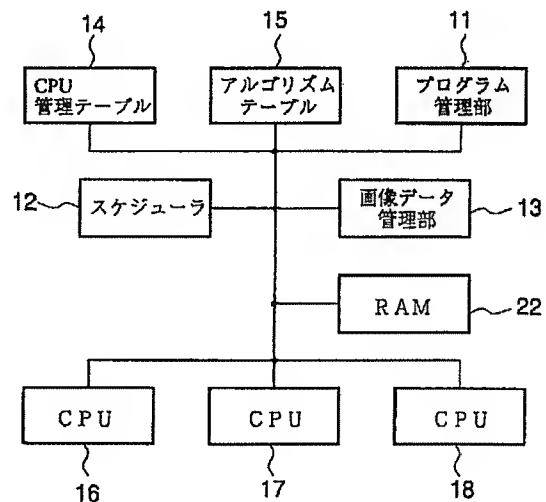
(74) 代理人 弁理士 大塚 康德 (外1名)

(54) 【発明の名称】 画像処理装置及びシステム

(57) 【要約】

【目的】 複数のCPUを備え、各CPUの負荷が動的に変化したり、各CPUの性能が異なったり、また、それらが組み合わされた状況においても、全体として最も処理効率が高くなるように画像データを分割して処理することができる画像処理装置及びシステムを提供する。

【構成】 CPU16~18の性能及び現在の負荷状況をCPU管理テーブル14で管理し、それより各CPUの予測効率を算出して、その結果に応じて画像データ管理部13で分割された画像データをスケジューラ12が各CPUに分配する。



【特許請求の範囲】

【請求項1】 複数のCPUを有する画像処理装置において、

各CPUの負荷の状況を判断する負荷判断手段と、
画像処理対象の画像データを保持する画像データ保持手段と、

前記画像データ保持手段に保持された画像データを複数の部分画像に分割し、前記負荷判断手段の判断結果に基づいて前記複数の部分画像のそれぞれを前記CPUに割り当てるデータ割り当て手段と、

各CPUにより処理された部分画像を統合して処理済み画像データとして出力する画像統合手段とを備えることを特徴とする画像処理装置。

【請求項2】 各CPUの性能を判断する性能判断手段を更に含み、

前記データ割り当て手段は前記負荷判断手段の判断結果と前記性能判断手段の判断結果とに基づいて各CPUにデータを割り当てることを特徴とする請求項1記載の画像処理装置。

【請求項3】 前記データ割り当て手段は画像データをCPUの数以上の部分画像に分割し、処理が終了したCPUに順次部分画像を表わす画像データを割り当てることを特徴とする請求項1記載の画像処理装置。

【請求項4】 前記データ割り当て手段により各CPUに割り当てられた部分画像の処理結果が画像データ全体に影響を及ぼす場合に、前記画像統合手段は各CPUの処理結果である全画像分の画像データを統合することを特徴とする請求項1記載の画像処理装置。

【請求項5】 前記データ割り当て手段は画像データの特徴により処理時間のかかる領域は小さく分割し、処理時間のかからない領域は大きく分割することを特徴とする請求項1記載の画像処理装置。

【請求項6】 複数のCPUを有する画像処理装置において、

各CPUの負荷の状況を判断する負荷判断手段と、
画像処理対象の画像データを保持するために各CPUに接続されたデータ保持手段と、

画像データを複数の部分画像に分割し、前記負荷判断手段の判断結果に基づいて前記複数の部分画像のそれぞれを各CPUのデータ保持手段に割り当てるデータ割り当て手段と、

部分画像毎の各CPUの処理結果を統合して1つの画像データとして出力する画像統合手段とを備えることを特徴とする画像処理装置。

【請求項7】 各CPUの性能を判断する性能判断手段を更に含み、

前記データ割り当て手段は前記負荷判断手段の判断結果と前記性能判断手段の判断結果とに基づいて各CPUにデータを割り当てることを特徴とする請求項6記載の画像処理装置。

【請求項8】 前記データ割り当て手段は画像データをCPUの数以上の部分画像に分割し、処理が終了したCPUに順次部分画像を表わす画像データを割り当てることを特徴とする請求項6記載の画像処理装置。

【請求項9】 前記データ割り当て手段により各CPUに割り当てられた部分画像の処理結果が画像データ全体に影響を及ぼす場合に、前記画像統合手段は各CPUの処理結果である全画像分の画像データを統合することを特徴とする請求項6記載の画像処理装置。

【請求項10】 前記データ割り当て手段は画像データの特徴により処理時間のかかる領域は小さく分割し、処理時間のかからない領域は大きく分割することを特徴とする請求項6記載の画像処理装置。

【請求項11】 複数のコンピュータを有する画像処理システムにおいて、

各コンピュータの負荷の状況を判断する負荷判断手段と、

画像データを複数の部分画像に分割し、前記負荷判断手段の判断結果に基づいて前記複数の部分画像のそれぞれを各コンピュータに割り当てるデータ割り当て手段と、
部分画像毎の各コンピュータの処理結果を統合して1つの画像データとして出力する画像統合手段とを備えることを特徴とする画像処理システム。

【請求項12】 各コンピュータの性能を判断する性能判断手段を更に含み、

前記データ割り当て手段は前記負荷判断手段の判断結果と前記性能判断手段の判断結果とに基づいて各コンピュータにデータを割り当てることを特徴とする請求項11記載の画像処理システム。

【請求項13】 前記データ割り当て手段は画像データをコンピュータの数以上の部分画像に分割し、処理が終了したコンピュータに順次部分画像を表わす画像データを割り当てることを特徴とする請求項11記載の画像処理システム。

【請求項14】 前記データ割り当て手段により各コンピュータに割り当てられた部分画像の処理結果が画像データ全体に影響を及ぼす場合に、前記画像統合手段は各コンピュータの処理結果である全画像分の画像データを統合することを特徴とする請求項11記載の画像処理システム。

【請求項15】 前記データ割り当て手段は画像データの特徴により処理時間のかかる領域は小さく分割し、処理時間のかからない領域は大きく分割することを特徴とする請求項11記載の画像処理システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は画像処理装置及びシステムに関し、特にマルチプロセッシングが可能な画像処理装置及びシステムに関する。

【0002】

【従来の技術】従来より、画像処理においては大量の画像データを高速に処理したいというニーズが大きい。従って、複数のCPUを1台の計算機上に載せた、SIMD (Single Instruction Multiple Data stream) 型やMIMD (Multiple Instruction Multiple Data stream) 型のコンピュータを使用して画像処理を行う技術が開発されてきた。SIMD型のコンピュータを使用する場合には、処理対象である画像データを複数領域に分割して各CPUに各領域毎の処理を割り振ることにより、複数領域に対して同時に同じ処理を行う。また、MIMD型のコンピュータを使用する場合には、処理対象である画像データを複数領域に分割してSIMD型のように各CPUに割りあてて処理したり、また、画像処理を各プロセス毎に、各CPUに割りあてることもある。

【0003】また、複数のコンピュータが高速のネットワークで接続されている場合には、ネットワークで接続された複数のコンピュータに対して画像処理を割りあてて分散処理を行うことにより、全体の処理速度の向上を図っている。

【0004】

【発明が解決しようとする課題】しかしながら、上述した従来のSIMD型コンピュータや、各CPUの役割が予め決められているMIMD型コンピュータにおいて、各CPUの性能は一定であり、各CPUにかかる負荷も予測できる。従って画像処理を行う場合に、処理効率が最適となるように、画像データの分割方法を予め決めておくことができる。しかし、他の処理が同時に実行されるようなMIMD型のコンピュータを使用する場合には、他の処理の実行状況によって画像処理の処理時間が変動するために、画像処理をするのに最も処理効率が良いCPUは、実際に処理を行う時点でないと決定できない。そのために、MIMD型のコンピュータを用いて複数のCPUに処理を分割させようとする場合、画像データの分割方法やCPUの割りあて方を固定としていたのでは、全体的な処理効率が最適とはならないという問題があった。

【0005】また、ネットワーク上に接続された複数のコンピュータに処理を分散させて実行効率を上げようとする場合においても、上述した場合と同様に実行時点での各コンピュータの状況及びネットワークの状況を判断して、画像データの分割方法や各処理装置の割りあてを決定しなければ、処理効率は良くならないという問題があった。

【0006】さらに、複数のCPUを備えるマルチCPUタイプのシステムにおいても、ネットワーク上に複数のコンピュータが接続されたシステムでも、処理実行途中に各CPUの負荷状況は変化してしまう可能性があり、従って処理実行前に各CPUに最適なジョブスケジューリングを行っても、実際に最適な処理効率が得られるとは限らないという問題があった。

【0007】従って本発明においては、上述した課題を解決するために、各CPUの負荷が動的に変化したり、各CPUの性能が異なったり、また、それらが組み合わされた状況においても、全体として最も処理効率が高くなるように画像データを分割して処理することができる画像処理装置及びシステムを提供することを目的とする。

【0008】

【課題を解決するための手段】上述した目的を達成するために、本発明は以下の構成を備える。

【0009】即ち、複数のCPUを有する画像処理装置において、各CPUの負荷の状況を判断する負荷判断手段と、画像処理対象の画像データを保持する画像データ保持手段と、前記画像データ保持手段に保持された画像データを複数の部分画像に分割し、前記負荷判断手段の判断結果に基づいて前記複数の部分画像のそれぞれを前記CPUに割り当てるデータ割り当て手段と、各CPUにより処理された部分画像を統合して処理済み画像データとして出力する画像統合手段とを備えることを特徴とする。

【0010】更に、各CPUの性能を判断する性能判断手段を更に含み、前記データ割り当て手段は前記負荷判断手段の判断結果と前記性能判断手段の判断結果とに基づいて各CPUにデータを割り当てることを特徴とする。

【0011】例えば、前記データ割り当て手段は画像データをCPUの数以上の部分画像に分割し、処理が終了したCPUに順次部分画像を表わす画像データを割り当てることを特徴とする。

【0012】例えば、前記データ割り当て手段により各CPUに割り当てられた部分画像の処理結果が画像データ全体に影響を及ぼす場合に、前記画像統合手段は各CPUの処理結果である全画像分の画像データを統合することを特徴とする。

【0013】例えば、前記データ割り当て手段は画像データの特徴により処理時間のかかる領域は小さく分割し、処理時間のかからない領域は大きく分割することを特徴とする。

【0014】また、複数のCPUを有する画像処理装置において、各CPUの負荷の状況を判断する負荷判断手段と、画像処理対象の画像データを保持するために各CPUに接続されたデータ保持手段と、画像データを複数の部分画像に分割し、前記負荷判断手段の判断結果に基づいて前記複数の部分画像のそれぞれを各CPUのデータ保持手段に割り当てるデータ割り当て手段と、部分画像毎の各CPUの処理結果を統合して1つの画像データとして出力する画像統合手段とを備えることを特徴とする。

【0015】

【作用】以上の構成により、各CPUの負荷が動的に変

化したり、各CPUの性能が異なったり、また、それらが組み合わされた状況においても、各CPUの予測効率を判断して全体で最も処理効率が高くなるように画像データを分割して処理することができるという特有の作用効果が得られる。

【0016】

【実施例】以下、本発明に係る一実施例について図面を参照して詳細に説明する。

【0017】<第1実施例>図1は、本実施例の画像処理装置の構成を示すブロック図である。図1において、1は画像データを入力するスキャナ、TVカメラ等の画像入力部、2は画像データを表示するCRT等の画像表示部、3は操作者が手操作入力を行うキーボード等の手操作入力部、4は画像を出力するプリンタ等の画像出力部、5は画像入力部1から入力された画像データに対して、後述する種々の画像処理を行って、画像表示部2や画像出力部4に出力する画像処理部である。尚、画像入力部1及び画像出力部4は他装置との通信による画像データの入出力を行ってもよい。

【0018】次に、上述した画像処理部5の詳細構成を図2のブロック図に示す。本実施例の画像処理装置はCPUを複数備えたマルチプロセッサ型となっている。図2において、16、17、18は実際に画像処理を実行するCPUである。11はRAM又はROM等により構成され、実行する画像処理プログラムを格納しているプログラム管理部、12は画像処理要求に従って、各CPU16~18に処理を分散して割り振る、いわゆるスケジューリングを行うスケジューラ、13は処理対象の画像データを管理する画像データ管理部、14は例えばCPU16~18等、本実施例装置において使用可能なCPUのアドレス及び性能、また、現在の負荷状況を管理しているCPU管理テーブルである。15は各種画像処理に関する基本的なアルゴリズム情報が関数として格納されているアルゴリズムテーブルである。22はRAMであり、各CPU16~18において作業領域として共有して使用される。図2に示す各構成は、バスにより接続されている。

【0019】尚、スケジューラ12は、CPU16~18のうちのいずれかにより、制御されている。また、プログラム管理部11に格納されている画像処理プログラムは操作者により作成され、アルゴリズムテーブル15に登録されている関数を利用して動作する。また、アルゴリズムテーブル15内の関数のアルゴリズムを変更して利用する場合には、変更したアルゴリズムの関数を作成し、アルゴリズムテーブル15に登録する。尚、アルゴリズムテーブル15に新たに関数に登録する場合、該関数が画像データをどのように分割して処理できるかの情報も登録しておく必要がある。

【0020】一般に画像処理においては、その処理のアルゴリズムによって画像データの分割方法が制限され

る。この画像データの分割処理は、例えば図3に示すように分類される。図3は、入力画像の画素範囲と、画像処理の結果の出力画像の画素範囲との関係を分類して示す図である。図3に示すように画像処理のタイプとしては、1画素を処理して1画素を得るタイプ、ある近傍領域を処理して1画素を得るタイプ、ある近傍領域を処理して近傍領域を得るタイプ、全体を処理して1画素を得るタイプ、全体を処理してある近傍領域を得るタイプ、全体を処理して全体を得るタイプ、1ラインを処理して1ラインを得るタイプ等がある。尚、画像全体を処理して全体を得る場合には、分割処理できないことは自明である。

【0021】次に、上述した図2に示すスケジューラ12の動作について説明する。スケジューラ12は、画像処理の要求が発生した場合に、使用可能な複数のCPU(CPU16~18)またはコンピュータに対して処理をそれぞれ分割して割り振り、全体としての処理効率を最大にすることを目的としている。

【0022】以降、複数のCPUへの処理の割り振り結果、即ち、どのCPUにどれだけの画像処理を分担させるかをスケジューラと称する。スケジューラ12においては、スケジューラを決定するための情報として、画像処理アルゴリズムテーブル15から画像処理で使用する各関数における画像データの分割方法、また、CPU管理テーブル14から各CPU16~18の性能を調べて、利用する。

【0023】スケジューラ12の動作を図4のフローチャートを参照して説明する。ステップS601において、スケジューラ12は画像処理プログラム管理部11からの処理要求を待ち、処理要求が発生すればステップS602に進み、アルゴリズムテーブル15を調べることによって、画像をどう分割できるかの情報を得る。次にステップS603に進み、CPU管理テーブル14に基づいて現在利用可能なCPU(例えば、CPU16~18)の数、及びそれぞれのCPUの性能を調べ、ステップS604でスケジューラは各CPUに現在の負荷の状態を問い合わせる。ここで、CPU16~18が利用可能であるとする。そして、スケジューラ12はステップS602~604で得られた情報を基に、ステップS605でどのCPUにどれだけの画像データを渡して処理させるか、即ちスケジューラを決定する。

【0024】本実施例においては、全画像データを後述する各CPUの予測効率の比に基づいて分割し、各CPUに割り当てる。従って、利用可能なCPUのそれぞれに割り当てられる画像データ量は異なっている。

【0025】スケジューラが決まれば、その後ステップS606において、各CPU16~18へ画像データの分割方法とその割当を通知する。次に、ステップS607において各CPU16~18へ画像処理コマンドを通知する。各CPU16~18では、スケジューラ12か

ら通知された画像処理コマンドに基づいて画像データ管理部13で管理されているRAM22上の画像データを処理し、処理が終了するとその旨のメッセージをスケジューラ12へ通知する。尚、この時、画像データ管理部13では、RAM22上において画像の同一領域を複数のCPUが同時にアクセスすることが無いように排他制御を行う。スケジューラ12は、各CPUが画像処理を実行している間は各CPUからの通知を待っているか、又は待ちながら別の処理を実行していてもよい。

【0026】ステップS608において、スケジューラ12はCPU16~18のうちの1つからの終了メッセージを受け取ると、ステップS609で処理を割り当てられたCPU16~18の全てから終了メッセージを受け取ったか否か、即ち、全CPUが処理終了したか否かを確認し、終了していなければステップS608へ戻り、他のCPUからのメッセージを待つ。全てのCPUの処理が終了したのであればステップS610へ進み、スケジューラ12はプログラム管理部11へ画像処理の終了を通知し、画像データ管理部13で各部分画像の処理結果をRAM22上で統合して1つの画像とするように依頼する。

【0027】以上説明したようにして、スケジューラ12では各CPUの画像処理のスケジュールを制御する。

【0028】次に、図2に示す画像データ管理部13について、詳細に説明する。画像データ管理部13は入力画像、及び処理後の画像を管理しており、スケジューラ12の指示に従って画像データの流れを制御する。ただし、画像データ管理部13は画像データの管理を行えばよく、画像データはプロセス内、RAM22内、その他のファイル等、どこに存在していてもよい。本実施例では、不図示のファイル内に画像データが格納されているとする。

【0029】RAM22も画像データ管理部13によって管理されており、画像データの流れは次のようになる。画像データ管理部13はプログラム管理部11からスケジューラ12へ要求した処理対象画像データをファイルI/Oの機能を用いて読み出し、RAM22上に展開する。この画像データをスケジューラ12の指示に従って分割し、各CPU16~18へ転送する。各CPUで処理が終了すると、画像データ管理部13はそれぞれのCPUから処理結果の画像データを回収して統合し、処理結果としてファイルへ書き出し、プログラム管理部11へ通知する。

【0030】本実施例においては、スケジューラ12により使用可能な複数のCPUに対して処理をそれぞれ分割して割り振るが、この時、各CPUの処理効率を予め予測することが必要となる。この予測されるCPUの処理効率を以下、予測効率と称し、本実施例における各CPUの予測効率の求めかた、及び画像の分割方法について説明する。

【0031】CPUの性能を s 、CPUの負荷を $load$ （100%の負荷で“1”となる）とすると、CPUの予測効率 pe は以下の式で表わされる。

$$【0032】pe = (1 - load) \times s$$

ここで、CPUの性能 s は実測値を基にして決定するのが最も良いが、例えばSPECint92（整数演算性能）や、SPECft92（浮動小数点演算性能）といったベンチマークプログラムによるCPUの性能の指標を基にして決めてもよい。また、CPUの負荷 $load$ は、CPUが現在どれだけ利用されているかを表すものであるため、CPUに実際に問い合わせることにより求められる。

【0033】また、画像の分割は、各CPUに対する処理の割り当てが予測効率 pe に比例するように決定する。ただし、画像によってはその処理の種類によって、分割できる場所が限定される場合がある。そのような場合には、なるべく予測効率と比例するように、分割可能な場所で分割するように調整する。

【0034】以下、本実施例における画像分割の例を示す。

【0035】例えば、6つのCPUが利用可能であり、それぞれの予測効率が1.0, 0.6, 0.5, 0.2, 0.2, 0.0であり、画像の大きさが512×512であり、処理はライン単位であってラインの途中で切断できないとする。この場合に、512ラインの画像を10:6:5:2:2に比例配分すると、それぞれ20, 12, 10, 4, 4, 0ラインの割り当てとなる。しかし、これでは2ライン余ってしまうため、実際には予測効率の最も高いCPUに余りを割り当て、それぞれ22, 12, 10, 4, 4, 0ラインに分割して、処理を行う。

【0036】一方、処理が画素単位の場合には、512×512を10:6:5:2:2に比例配分し、余りを予測効率に応じて分配して割り当てると、それぞれ104860, 62914, 52428, 20971, 20971が割り当てられる。

【0037】以上説明したように本実施例によれば、複数のCPUを備えるシステムにおいて、各CPUが他のジョブを同時に処理するために動的に負荷が変化する状況や、各CPUの性能が異なる状況等においても、各CPUの予測効率を判断することにより、最も効率良く画像処理を行うことが可能となる。

【0038】<第2実施例>以下、本発明に係る第2実施例について、図面を参照して説明する。

【0039】第2実施例における画像処理装置の構成は上述した第1実施例に示す図1と同様であるが、画像処理部5の詳細構成が第1実施例と異なる。第2実施例における画像処理部5の詳細構成を図5のブロック図に示す。

【0040】図5において、上述した図2と同様の構成

には同一番号を付し、説明を省略する。図5において、CPU16~18は、それぞれ固有のRAM23、24、25を備える。

【0041】以下、第2実施例におけるスケジューラ12の処理を、図6のフローチャートを参照して説明する。

【0042】図6は第2実施例のスケジューラ12の動作を説明するフローチャートであり、ステップS701~ステップS705までは上述した第1実施例の図4に示すステップS601~ステップS605と同様の処理であるため、説明を省略する。

【0043】ステップS706において、スケジューラ12は画像データの分割方法をCPUではなく、画像データ管理部13へ送る。そして、ステップS707でスケジューラ12は画像データ管理部13に対して、各CPU16~18へ分割した画像データを転送するように要求を出す。そしてステップS708において、各CPU16~18に対して画像処理コマンドを発行する。すると各CPU16~18では、スケジューラ12から通知された画像処理コマンドに基づいて、画像データ管理部13から送られた画像データそれぞれのRAM23~25上で処理し、処理が終了するとその旨のメッセージをスケジューラ12へ通知する。また、処理された画像データは、画像データ管理部13へ送られる。

【0044】そしてステップS709において、スケジューラ12はCPU16~18のうちの1つからの終了メッセージを受け取ると、ステップS710で処理を割り当てられたCPU16~18の全てから終了メッセージを受け取ったか否か、即ち、全CPUが処理終了したか否かを確認し、終了していなければステップS709へ戻り、他のCPUからのメッセージを待つ。全てのCPUにおいて処理が終了したのであればステップS711へ進み、プログラム管理部11へ画像処理の終了を通知し、画像データ管理部13へ各部分画像の処理結果の統合を依頼する。

【0045】以上説明したようにして、スケジューラ12では各CPUの画像処理のスケジュールを制御する。

【0046】即ち、第2実施例においては、スケジューラ12において画像データの分割方法を画像データ管理部13へ送り、画像データ管理部13では、その分割方法に従って分割した画像データを各CPUへ転送することにより、各CPUの作業領域が共通でない場合にも、効率良く画像処理を行うことができる。

【0047】また、第2実施例における方法は、図5に示す複数のCPUを備える画像処理装置のみでなく、例えば図7に示すように、複数のコンピュータ41~44をネットワーク45で接続した画像処理システムにおいても適用可能である。この場合、コンピュータ41~44のうちのいずれかが図5に示すCPU16~18及びRAM23~25以外の構成を備えており、そのコンピ

ュータにより第2実施例における画像処理が統括される。

【0048】<第3実施例>以下、本発明に係る第3実施例について、図面を参照して説明する。

【0049】第3実施例における画像処理装置の構成及び画像処理部5の詳細構成は、上述した第1実施例に示す図1及び図2と同様であるため、説明を省略する。

【0050】以下、第3実施例におけるスケジューラ12の処理を、図8のフローチャートを参照して説明する。

【0051】図8は第3実施例のスケジューラ12の動作を説明するフローチャートであり、ステップS801~ステップS805までは上述した第1実施例の図4に示すステップS601~ステップS605と同様の処理であるため、説明を省略する。

【0052】ステップS806において、スケジューラ12は各CPUへ画像処理コマンド及び処理する部分画像の情報を送る。そして、ステップS807で、処理を終了したCPUが現れるのを待つ。処理を終了したCPUが現れるとステップS808に進み、全部分画像について処理が終了したか否かを判定する。未処理の部分画像があればステップS809に進み、ステップS807で処理終了を確認されたCPUに未処理の部分画像情報を転送する。一方、ステップS808において全部分画像、即ち全画像領域について処理が終了したと判定されると、ステップS810に進んで処理結果の画像データを画像データ管理部13で統合するよう指示し、プログラム管理部11へ通知する。

【0053】以上説明したようにして、スケジューラ12では各CPUの画像処理のスケジュールを制御する。

【0054】例えば、利用可能なCPU数が5である場合には、画像データをCPUの数5より大きな数の同じ大きさの部分画像に分割する。例えば、画像データを15個の部分画像に分割する。そして、まずそのうちの5個の部分画像をそれぞれのCPUへ割り当てる。各CPUの負荷の状況や性能の差により、処理が終了する時間はそれぞれ異なるため、第3実施例においては早く終了したCPUに、順次他の部分画像を転送して処理を行う。このとき、利用可能なCPUの数と画像データを分割する数の関係はあらかじめ決めておいてもよいし、何らかの評価関数を用いて、画像処理装置内で自動的に決定されるようにしてもよい。

【0055】また、第3実施例は図7に示す複数台のコンピュータを接続したネットワークシステムにおいても適用可能である。この場合、CPUの数に対する画像データの分割数の関係が、通信時間とも関係してくる。ネットワークシステムにおいては処理時間全体に占める通信時間の割合が大きいため、処理する画像データの分割数を大きくすると、各コンピュータへ通信時間がかかってしまい効率的ではない。しかし、画像データの分割数

を小さくすると、一部のコンピュータのみに負荷が集中してしまう可能性がある。そのため、例えば利用可能なCPUの数の2倍とか3倍といったように、画像データの分割数にはある程度の範囲を設定すると良い。

【0056】尚、第3実施例におけるスケジューラの処理は、上述した図5に示す構成でももちろん適用可能である。

【0057】以上説明したように第3実施例によれば、各CPUの処理途中でその負荷が大きく変化するような場合でも、処理可能なCPUの数以上に入力画像を分割し、処理が終了したCPUに順次部分領域を割り当てることにより、負荷の変化に関らず処理効率を高めることができる。

【0058】＜第4実施例＞以下、本発明に係る第4実施例について、図面を参照して説明する。

【0059】第4実施例における画像処理装置の構成及び画像処理部5の詳細構成は、上述した第2実施例に示す図1及び図5と同様であるため、説明を省略する。

【0060】第4実施例においては、画像の部分領域を対象とした処理が全体の領域に影響を及ぼす場合の例を示す。

【0061】画像の部分領域を対象とした処理が全体の領域に影響を及ぼす場合の例として、例えば画像中から直線等、パラメータで表現できる図形を抽出するハフ(Hough)変換について考える。ハフ変換は、入力画像の各画素毎に処理を実行し、結果のパラメータ空間の2次元配列上の何処かのセルの値を変化させる。以下、ハフ変換の処理例を図9を参照して説明する。ハフ変換は、図9の(a)に示すように、点A、B、C、Dを含む直線 α を

$$\rho = x \cdot \cos \theta + y \cdot \sin \theta \quad \dots \quad (式1)$$

で表現したときに、x軸、y軸の2次元で表せる画像を、図9の(b)に示す ρ 軸、 θ 軸の2次元のパラメータ空間上の正弦曲線に変換するものである。尚、図9の(a)において ρ は原点から直線へ降ろした垂線の長さ、 θ は垂線とx軸とのなす角である。即ち、例えば図9の(a)に示す点Aを通る全ての直線群が、図9の(b)に示す曲線Aで表わされる。そして、図9の(a)に示されるxy空間の各点(例えば、A、B、C、D)について、式1を満足するような図9の(b)に示す(ρ 、 θ)の曲線(例えばA、B、C、D)を計算し、 $\rho\theta$ 空間上の対応する曲線をカウントアップしていく。そして最終的に、図9の(b)に示す $\rho\theta$ 空間でのピーク α を検出することにより、そのピーク α を通る(ρ 、 θ)パラメータ空間上の曲線A、B、C、Dに対応する、xy空間中の点A、B、C、Dを通る直線 α が存在することがわかる。即ちハフ変換においては、xy空間の直線が $\rho\theta$ 空間上の1点で表現される。

【0062】以上説明したような処理の場合、入力画像を適当な方法で分割して各CPUに割り当てることは可

能であるが、同時に処理を行う全てのCPUが、処理結果である $\rho\theta$ 空間を格納するための、同サイズの画像データ領域をそれぞれ独立して持つことが必要である。

【0063】これに対して、上述した第1実施例～第3実施例においては、入力画像内における処理対象の部分領域の位置と、その部分領域を処理した結果の出力画像内における位置とは一致していた。従って、入力画像を分割した方法と全く逆に処理結果を統合すれば、出力画像を得ることができた。

【0064】しかしながら第4実施例においては、同じ座標に対する各CPUの処理結果をそれぞれ加算するという画像の統合を行う必要がある。この様子を図10、図11に示す。図10は、上述した第1実施例～第3実施例における画像の統合方法を示しており、入力画像、出力画像ともに同じ位置、同じ大きさで画像が分割される。一方、図11は第4実施例における画像の統合方法を示しており、入力画像の分割方法に関わらず、各CPUは同じ大きさの画像データ領域(P1、P2、P3)を有しており、そこに処理結果を格納する。そして、各CPUの画像データ領域(P1、P2、P3)において、その同じ位置の処理結果を加算することにより、出力結果を得る。

【0065】従って第4実施例においては、入力画像の各部分領域を処理した結果が出力画像の全体に影響するような場合でも、入力画像を分割して複数のCPUに割り当てて、その結果を上記説明したように統合することによって、効率良く画像処理を行うことができる。

【0066】尚、第4実施例はハフ変換を例として説明を行ったが、本実施例はこの例に限定されるものではなく、パラメータ変換等、部分画像の処理結果が画像全体に影響を及ぼすような全ての画像処理に対して、適用可能である。

【0067】また、例えば図2に示すように複数のCPUが1つの共有メモリを有する場合には、共有メモリ上に結果の配列の領域を確保し、複数のCPUからアクセスすることにより、処理を行えばよいが、但しこの場合には、同じアドレスを同時に複数のCPUがアクセスすることがないように、排他制御を行う必要がある。

【0068】＜第5実施例＞以下、本発明に係る第5実施例について、図面を参照して説明する。

【0069】第5実施例における画像処理装置の構成及び画像処理部5の詳細構成は、上述した第2実施例に示す図1及び図5と同様であるため、説明を省略する。

【0070】第5実施例においては、画像の全体領域を処理対象とし、部分領域の性質によって処理速度が異なるような画像処理を行う場合の例を示す。このような処理としては、例えば、ラベリングした領域の特徴量の計測、特徴点のマッチング、領域分割等がある。

【0071】ラベリングした領域の特徴量の計測において、処理に要する時間は、部分画像中に存在するラベル

リングされた領域数に依存する。従って、各部分画像に含まれるラベリング領域数が一定になるように画像データを分割することにより、画像全体の処理効率は向上する。

【0072】図12に、黒く塗りつぶした領域の特徴量を計算する例を示す。このような場合には、例えば図12の破線の下上の領域に入力画像を分割すると、各領域とも6個のラベリング領域を有することにより、2つのCPUに負荷を均等に分割できる。また、各CPUの予測効率を調べ、その予測効率に合わせて分割位置を変更することによって、予測効率の高いCPUにはラベル領域の多い領域を、予測効率の低いCPUにはラベル領域の少ない領域を割り当てれば良い。こうすることにより、全体としての処理効率の向上を図ることができる。

【0073】次に、特徴点のマッチング処理について説明する。特徴点のマッチングとは、なんらかの方法で画像中の特徴点を抽出し、それぞれの特徴点が他の画像のどの画素位置に対応するかを調べる処理である。この処理は、例えば3次元座標情報を抽出可能なステレオ画像の対応点を抽出して、特徴点での奥行き情報を得る場合等に重要な処理である。このような場合には、全体の処理時間は特徴点の数に依存する。そのため、特徴点の数が一定になるように、画像を部分領域に分割して各CPUに割り当てることにより、全体の処理効率は向上する。また、特徴点に対応する位置を調べる必要がある領域は、撮影条件による拘束からある程度限定することができる。従って、例えば図2又は図7に示すように各CPU又はコンピュータが共有領域を持たない場合には、全ての画像を各CPUに送る必要はなく、必要な部分だけを送ることにより、画像データの転送によるオーバーヘッドを少なくすることが可能である。

【0074】また、領域分割処理においては、近傍の画素の輝度値を比較し、その差が所定値未満である場合に、それらの画素は同一の領域にあると見なす。そして、この処理をその輝度値の差が所定値以上であるような画素がなくなるまで繰り返す。また、各領域の大きさを計測して、各領域の大きさが一定以上になるように領域の統合処理を行う場合もある。

【0075】このような分割処理においては、分割する画像の特徴に応じて、予めどの程度の繰り返して処理が収束するかを予測できることがある。例えば、初めから同一輝度値の領域に関しては領域の統合処理は不要であり、また、同一輝度値でなくても輝度値の差が小さいことから予め分かっている領域については、収束が速いことが予測される。このような統計的な手法を利用することによって、予め収束の速い領域と、収束の遅い領域とを区別することが可能である。そして、この情報を利用して画像の分割方法を決定することができる。

【0076】以上説明したように第5実施例によれば、画像の全体領域を処理対象とし、部分領域の性質によ

って処理速度が異なるような画像処理を行う場合においても、複数のCPUに画像を効率良く割り当てることにより、効率良く画像処理を行うことができる。

【0077】また、第5実施例における画像処理装置は、図2、又は図7に示すように構成でも実現可能である。

【0078】尚、本発明は、複数の機器から構成されるシステムに適用しても1つの機器から成る装置に適用しても良い。また、本発明は、システム或は装置にプログラムを供給することによって達成される場合にも適用できることはいうまでもない。

【0079】

【発明の効果】以上説明したように、本発明によれば、複数のCPUからなり、処理を各CPUに分割して割り当てることによって画像処理を行う画像処理システムにおいて、各CPUの負荷が動的に変化する状況や、各CPUの性能が異なる状況、またそれらが組み合わせられた状況においても、処理を実行する時点で、各CPUの予測効率を動的に判断しながら、全体で最も処理効率が高くなるように入力画像を分割して画像処理を行うことが可能となる。

【0080】更に、処理途中で負荷が大きく変化するような場合には、処理可能なCPUの数以上に入力画像を分割し、処理が終了したCPUに順次部分領域を割り当てることにより、負荷の変化に関らず処理効率を高めることができる。

【0081】また、入力画像の各部分領域を処理した結果が出力画像の全体に影響するような場合でも、入力画像を分割して複数のCPUに割り当てて、その結果を全て統合することによって、効率良く画像処理を行うことができる。

【0082】更に、入力画像の部分領域の性質によって処理速度が異なるような処理をする場合には、部分領域の性質に従って分割方法を変えることによって、各CPUの処理時間の差が少なくなるように分割することが可能となり、全体としての処理効率の向上を図ることができる。

【0083】

【図面の簡単な説明】

【図1】本発明に係る第1実施例における画像処理装置の構成を示すブロック図である。

【図2】本実施例における画像処理部の詳細構成を示すブロック図である。

【図3】本実施例における入力画像の画素範囲と画像処理の結果の出力画像の画素範囲との関係を分類して示す図である。

【図4】本実施例におけるスケジューラの動作を説明するフローチャートである。

【図5】本発明に係る第2実施例における画像処理部の詳細構成を示すブロック図である。

【図6】本発明に係る第2実施例におけるスケジューラの動作を説明するフローチャートである。

【図7】本発明に係る第2実施例における画像処理ネットワーク構成を示す図である。

【図8】本発明に係る第3実施例におけるスケジューラの動作を説明するフローチャートである。

【図9】本発明に係る第4実施例におけるハフ変換を説明するための図である。

【図10】本発明に係る第1実施例から第3実施例までの処理における入力画像と出力画像との分割領域の対応を示す図である。

【図11】本発明に係る第4実施例における入力画像と出力画像との対応を示す図である。

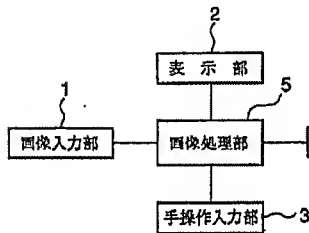
【図12】本発明に係る第5実施例におけるラベリング

された入力画像の分割例を示す図である。

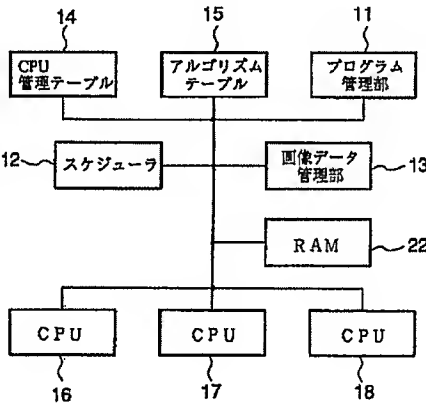
【符号の説明】

- 1 画像入力部
- 2 表示部
- 3 手操作入力部
- 4 画像出力部
- 5 画像処理部
- 11 プログラム管理部
- 12 スケジューラ
- 13 画像データ管理部
- 14 CPU管理テーブル
- 15 アルゴリズムテーブル
- 16, 17, 18 CPU
- 22 RAM

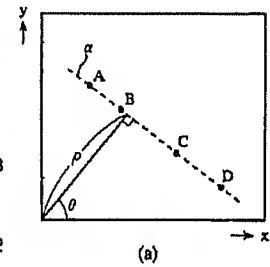
【図1】



【図2】



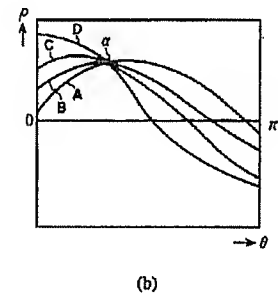
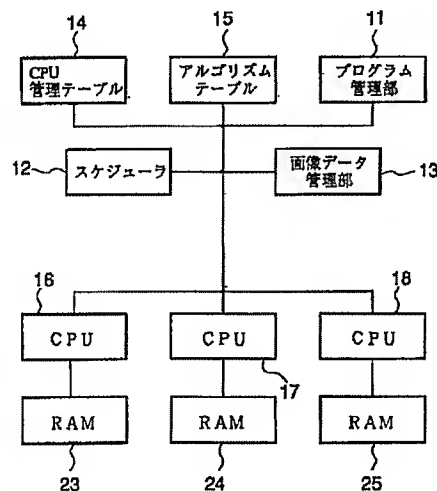
【図9】



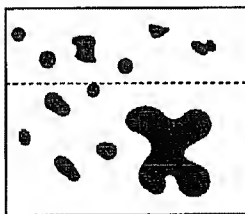
【図3】

入力画像	出力画像
1画素	1画素
近傍	1画素
近傍	近傍
全体	1画素
全体	近傍
全体	全体
ライン	ライン

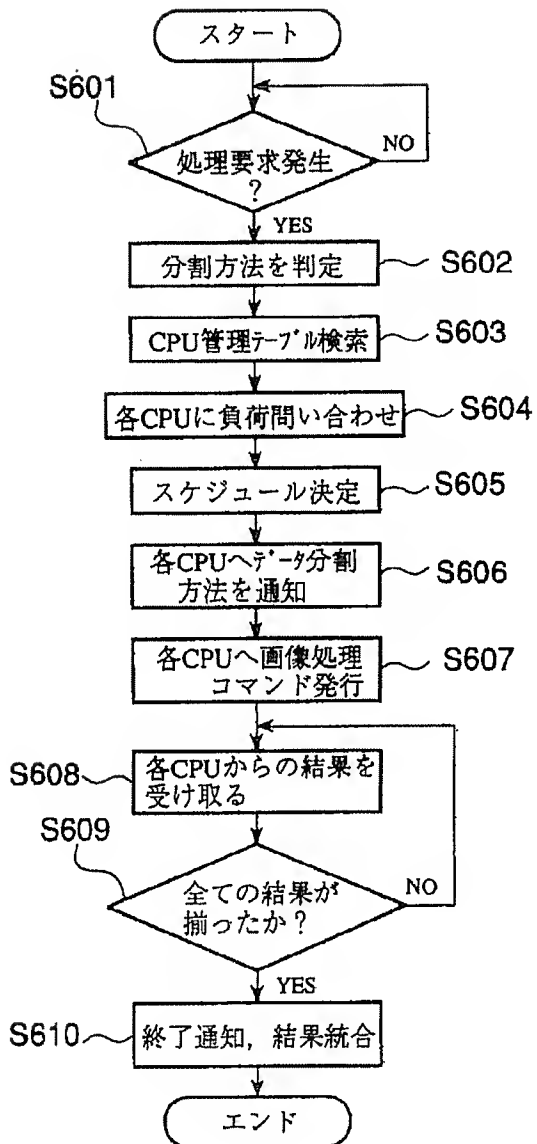
【図5】



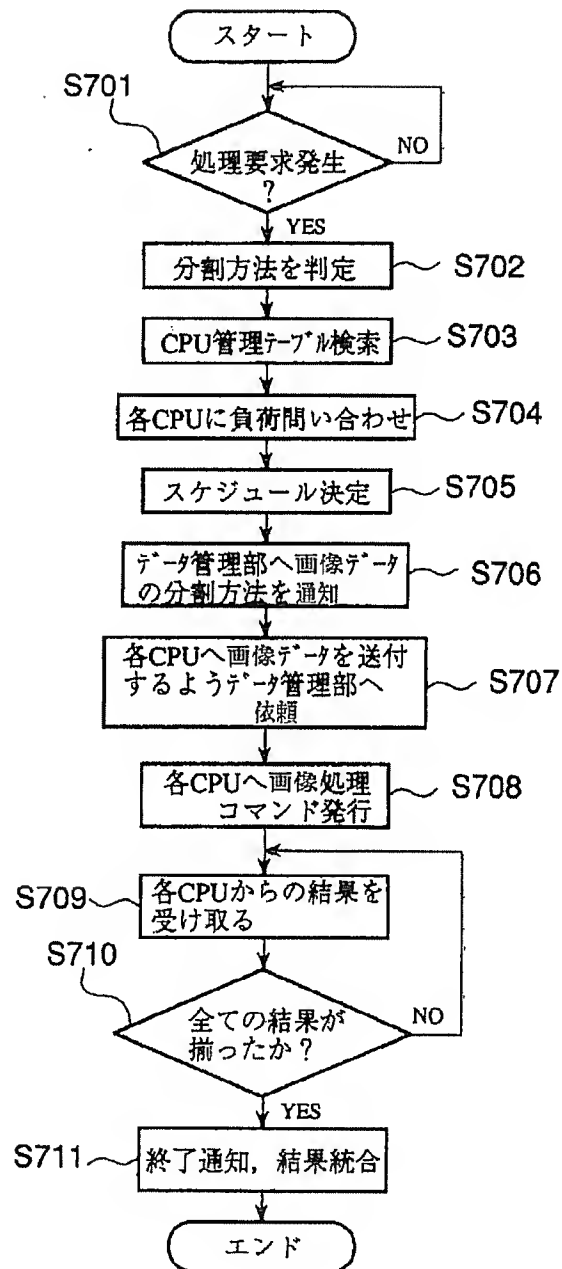
【図12】



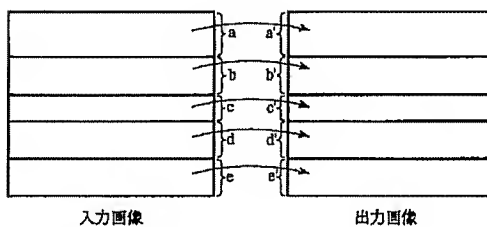
【図4】



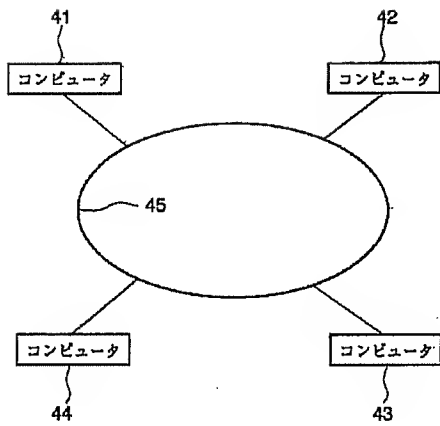
【図6】



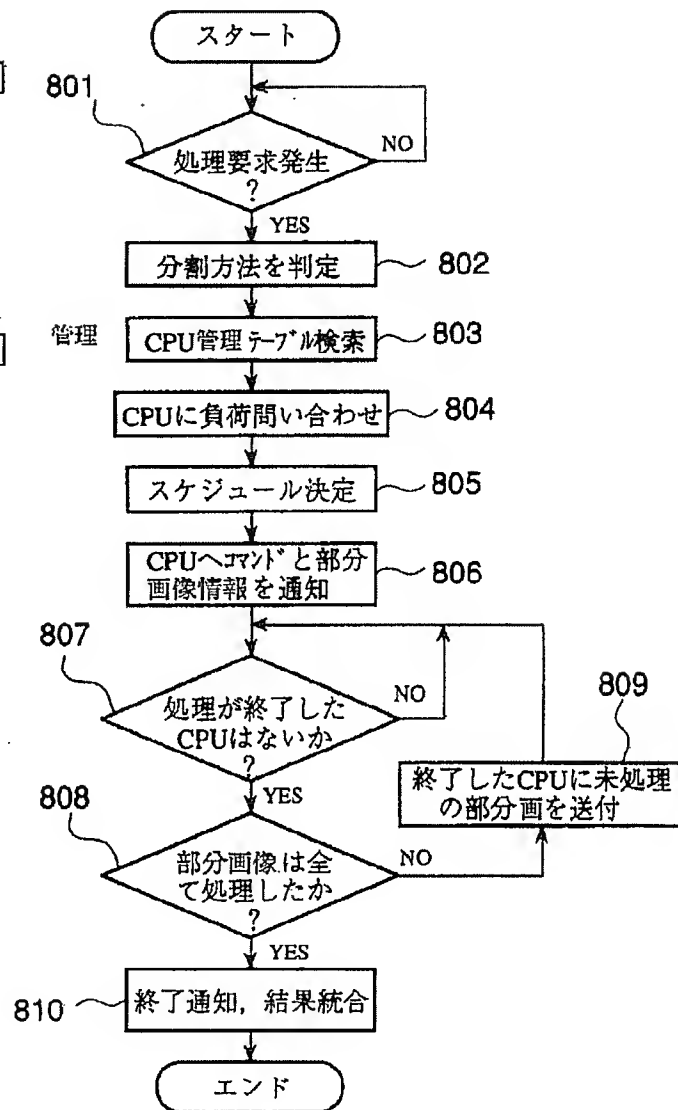
【図10】



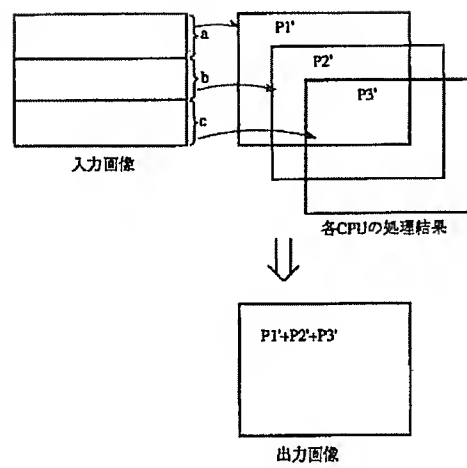
【図7】



【図8】



【図11】



フロントページの続き(51) Int. Cl.⁶

G 0 6 T 1/20

// G 0 6 T 7/00

識別記号

庁内整理番号

F I

技術表示箇所